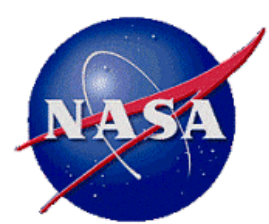


# **Using CMMI for Improvement at GSFC**

**Systems Engineering Lecture Series  
6/01/04**

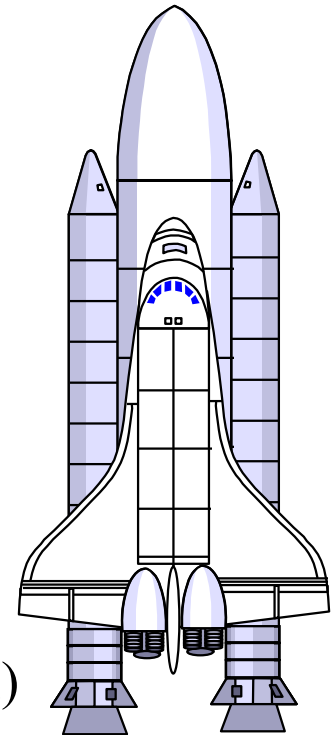
**Sally Godfrey  
Sara.H.Godfrey@nasa.gov  
301-286-5706**

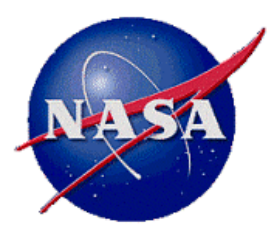


# Agenda

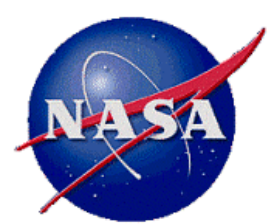


- CMMI: What is it? Why use it?
- NASA Improvement Initiatives
  - Systems Engineering & CMMI
  - Software Engineering & CMMI
- GSFC's Use of CMMI for Software
  - Phase 1: Piloting
    - What we learned during piloting (FY02)
  - Phase 2: Implementation
    - Approach for implementing improvement (CMMI)
    - Progress to date
- Summary





# What is CMMI?



# What is CMMI?

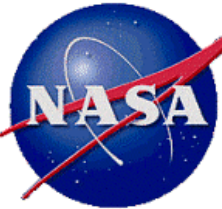
The Capability Maturity Model Integrated (CMMI) is an **integrated framework for maturity models** and associated products that integrates the two key disciplines that are inseparable in a systems development activity: software engineering and systems engineering.

A **common-sense application** of process management and quality improvement concepts to product development, maintenance and acquisition

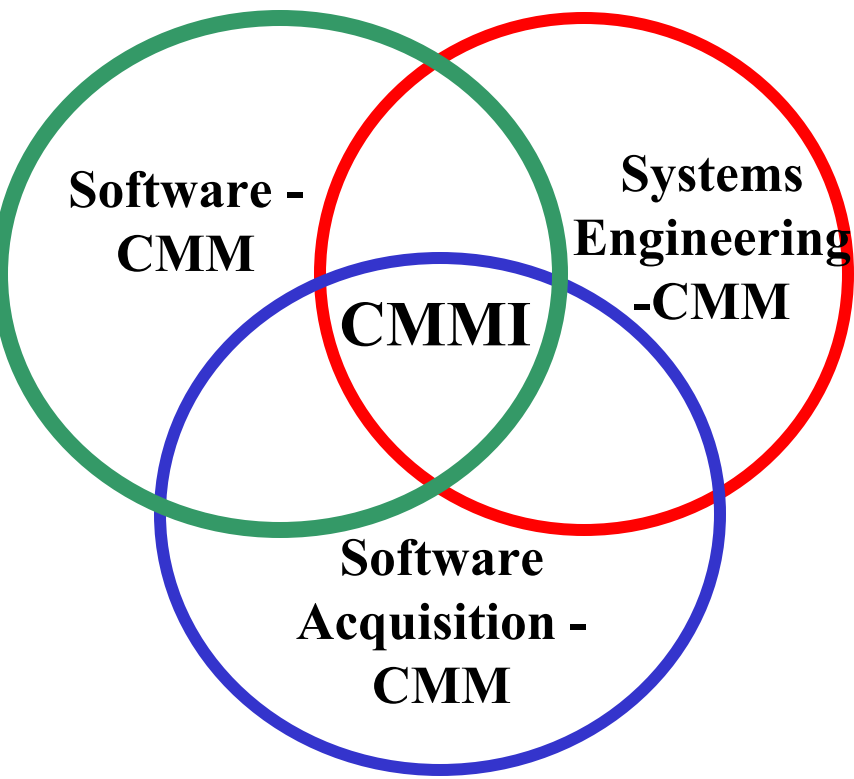
A **set of best practices**

A **community developed guide**

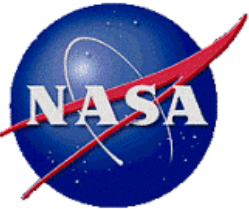
A **model for organizational improvement**



# Capability Maturity Model Integrated (CMMI)-Staged



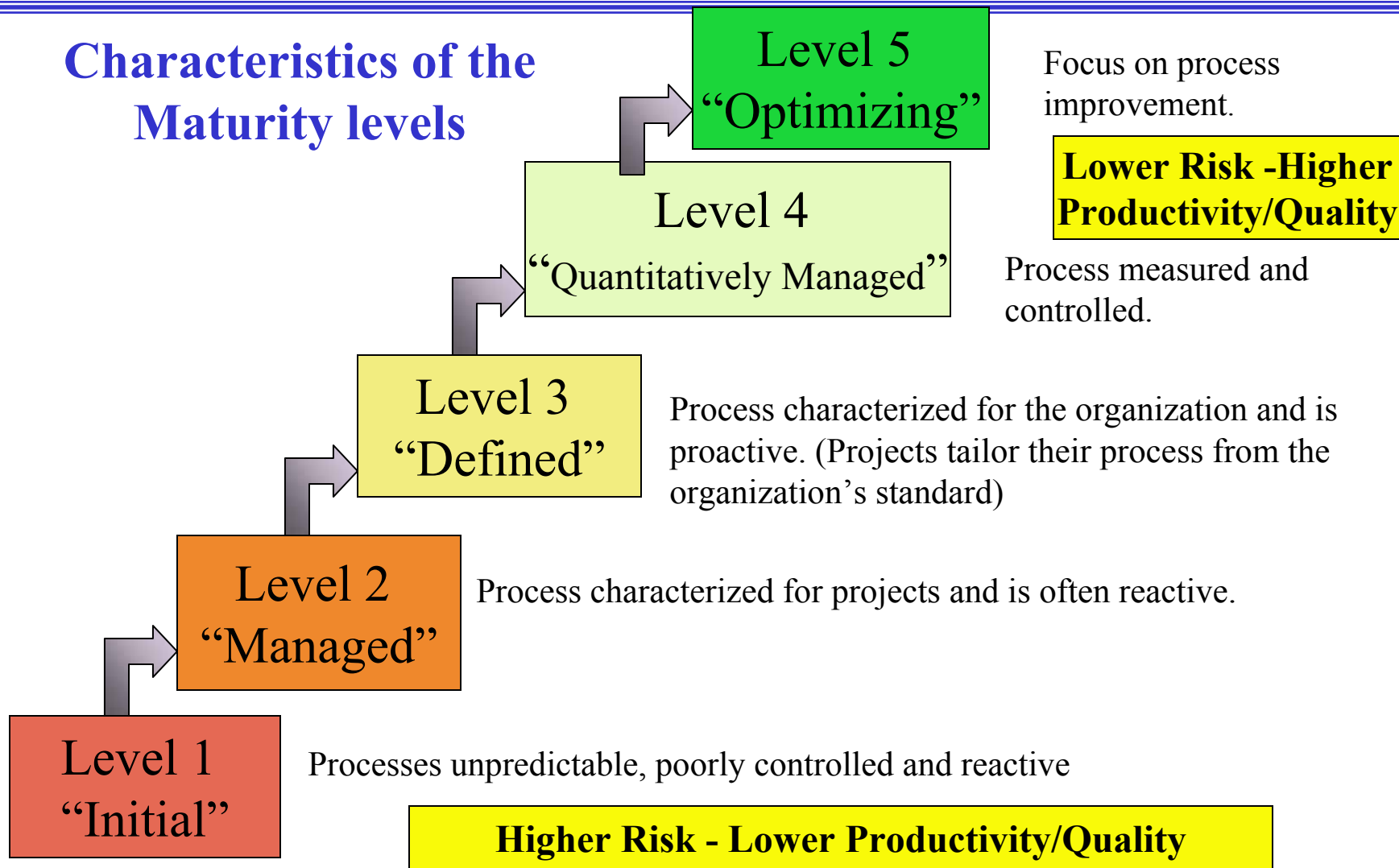
Level	Process Areas
5 Optimizing	<b>Organization innovation and deployment</b> <b>Causal analysis and resolution</b>
4 Quantitatively Managed	<b>Organizational process performance</b> <b>Quantitative project management</b>
3 Defined	<b>Requirements development</b> <b>Technical solution</b> <b>Product integration</b> <b>Verification</b> <b>Validation</b> <b>Organizational process focus</b> <b>Organizational process definition</b> <b>Organizational training</b> <b>Integrated project management</b> <b>Risk management</b> <b>Decision analysis and resolution</b> <b>Integrated Supplier Management</b> <b>Integrated Teaming</b>
2 Managed	<b>Requirements management</b> <b>Project planning</b> <b>Project monitoring and control</b> <b>Configuration Management</b> <b>Supplier agreement management</b> <b>Measurement and analysis</b> <b>Product &amp; Process Quality Assurance</b>
1 Initial	



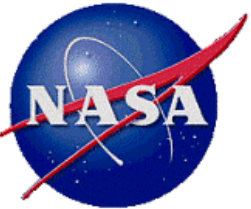
# Capability Maturity Model Integrated -Staged



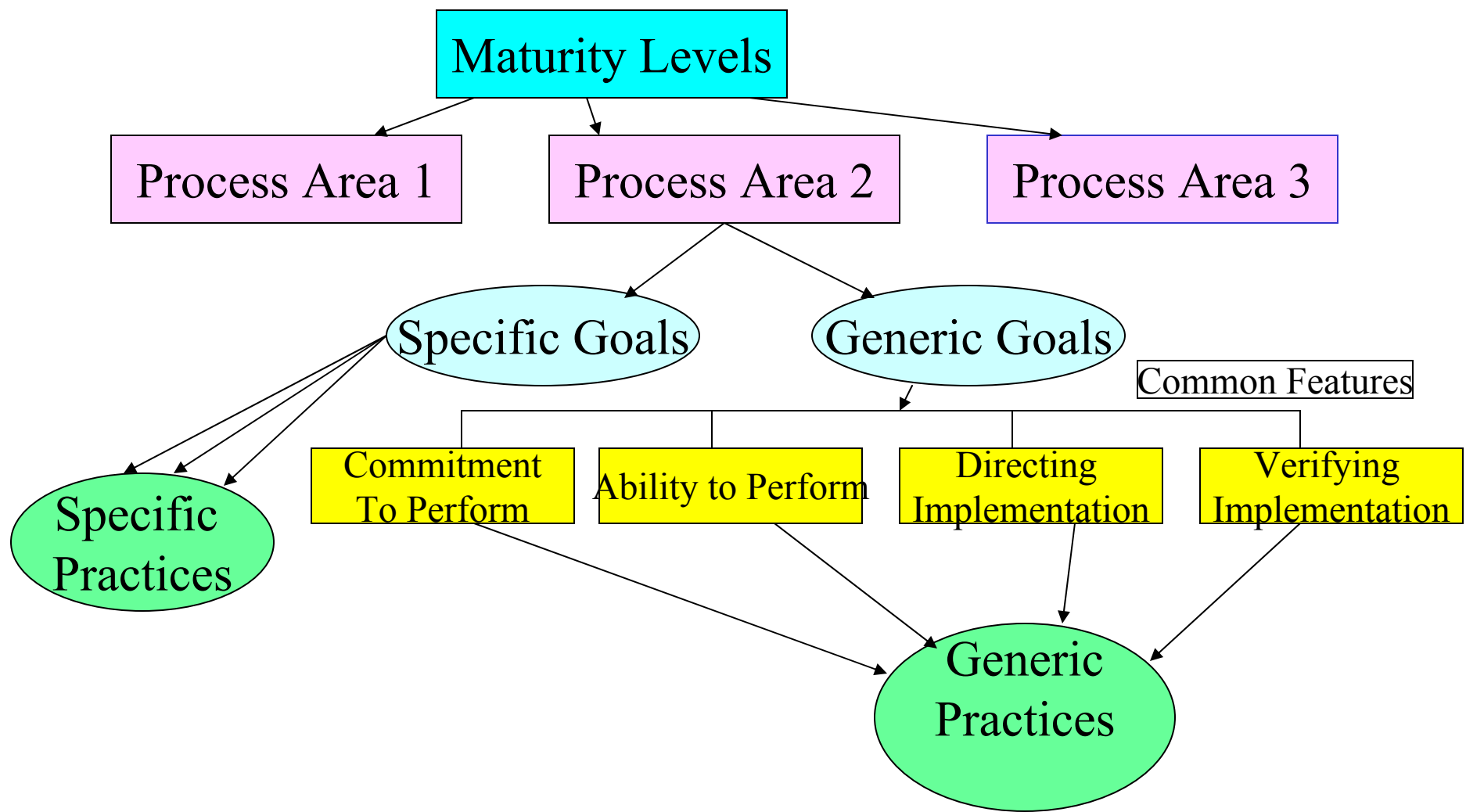
## Characteristics of the Maturity levels

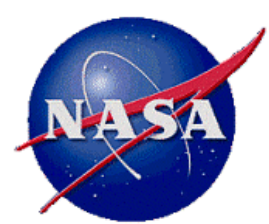


CMM was developed by the Software Engineering Institute (SEI), Carnegie Mellon University (CMU)



# Components of CMMI Model





# Example Process Area: Requirements Management



## SG 1: Manage Requirements

SP 1.1: Obtain an Understanding of the Requirements

SP1.2: Obtain Commitment to the Requirements

SP1.3: Manage Requirements Changes

SP1.4: Maintain Bi-directional Traceability of Requirements

SP1.5: Identify Inconsistencies between Project Work & Reqmts

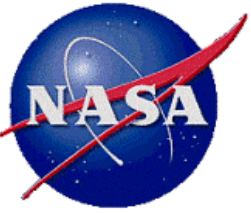
## GG 2: Institutionalize a Managed Process

GP 2.1: Establish an Organizational Policy

GP 2.2: Plan the Process

GP 2.3: Provide Resources

GP 2.4: Assign Responsibility



# Example Process Area: Requirements Management



## GG 2: Institutionalize a Managed Process

GP 2.5: Train People

GP 2.6: Manage Configurations

GP 2.7: Identify & Involve Relevant Stakeholders

GP 2.8: Monitor and Control the Process

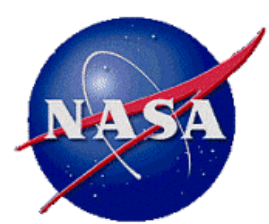
GP 2.9: Objectively Evaluate Adherence

GP 2.10: Review Status with Higher Level Management

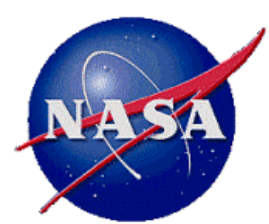
## GG 3: Define a Managed Process

GP 3.1: Establish a Defined Process

GP 3.2: Collect Improvement Information

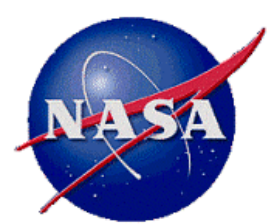


# Why are we using CMMI?



# Why Use CMMI?

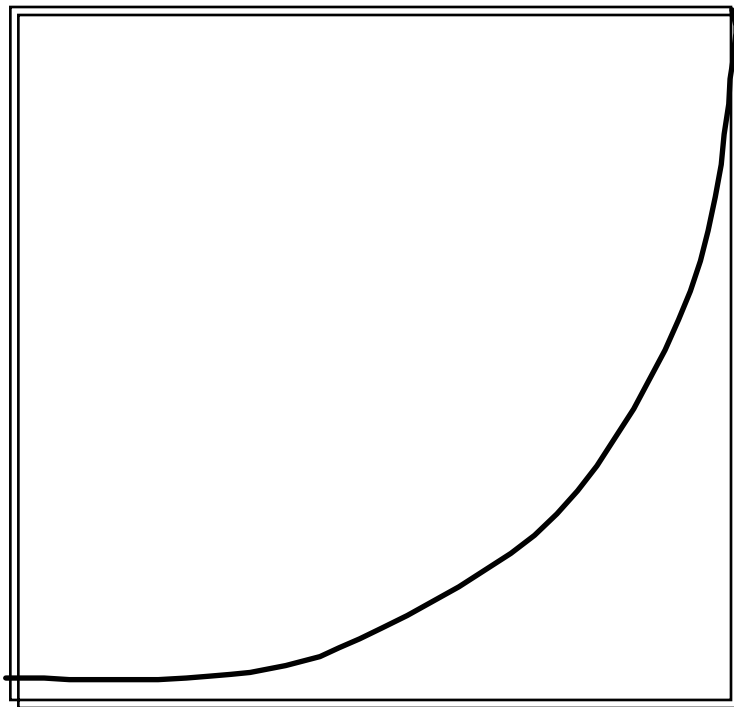
- In software and systems engineering, it is a **benchmarking tool** widely used by industry and government, both in the US and abroad.
- CMMI acts as a **roadmap** for process improvement activities.
- It provides **criteria** for reviews and appraisals.
- It provides a **reference point** to establish present state of processes.
- CMMI addresses practices that are the **framework** for process improvement.
- CMMI is **not prescriptive**; it does not tell an organization how to improve.



# Growth Trend Problem: Dependency on Software Technology



Increasing amount of project software



Years →

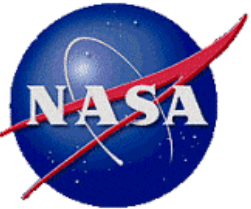
**Indicator:** Industry has reported that the amount of software on passenger aircraft is increasing exponentially

NASA programs and projects are likely to be experiencing the same growth curve

The use of software as a technology is on a much steeper growth curve than other supporting technologies

If the Agency does nothing to improve software engineering and acquisition, we can expect commensurate growth in cost, schedule, and defects

Uncontrolled growth of software dependencies without prudent mitigations will result in a real reductions in NASA's capability to fulfill it's mission

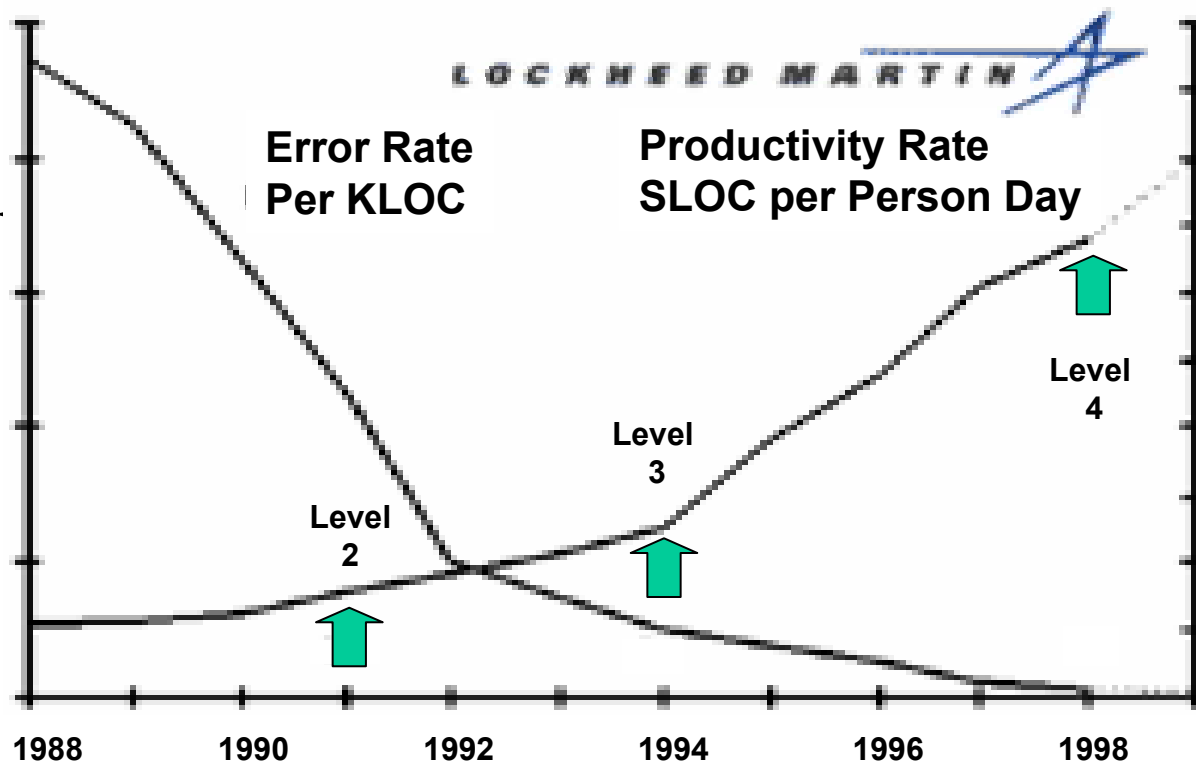


# Improvements with CMM:Time History - Productivity/Error Rates



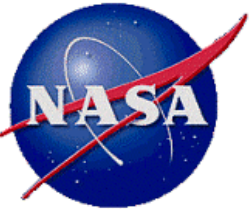
## Productivity Rate and Quality Performance

\* For Software Programs



**Productivity  
Increased By  
80% As Error  
Rates  
Decreased**

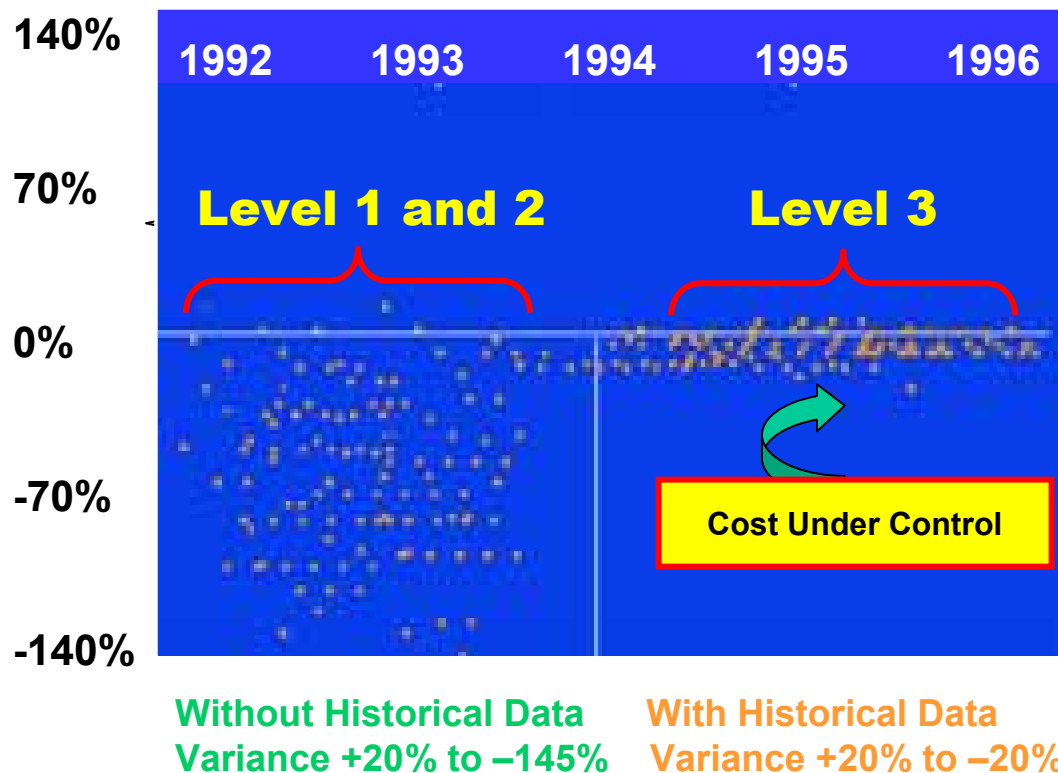
Source: Lockheed Martin SEPG Presentation 1999



# Improvements with CMM: Time History – Cost



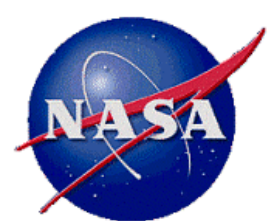
## Project Cost Estimates Labor Hours Over- or Under-Estimated



**Product  
Quality  
Increased with  
Rising Maturity**

**Based on 120 Projects in  
Boeing Information  
Systems**

Reference: Scott Griffin, Chief Information Officer, The Boeing Company, SEPG Conference, 2000.

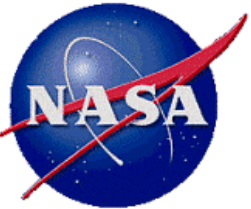


# Project Performance vs. CMM Level (General Dynamics)

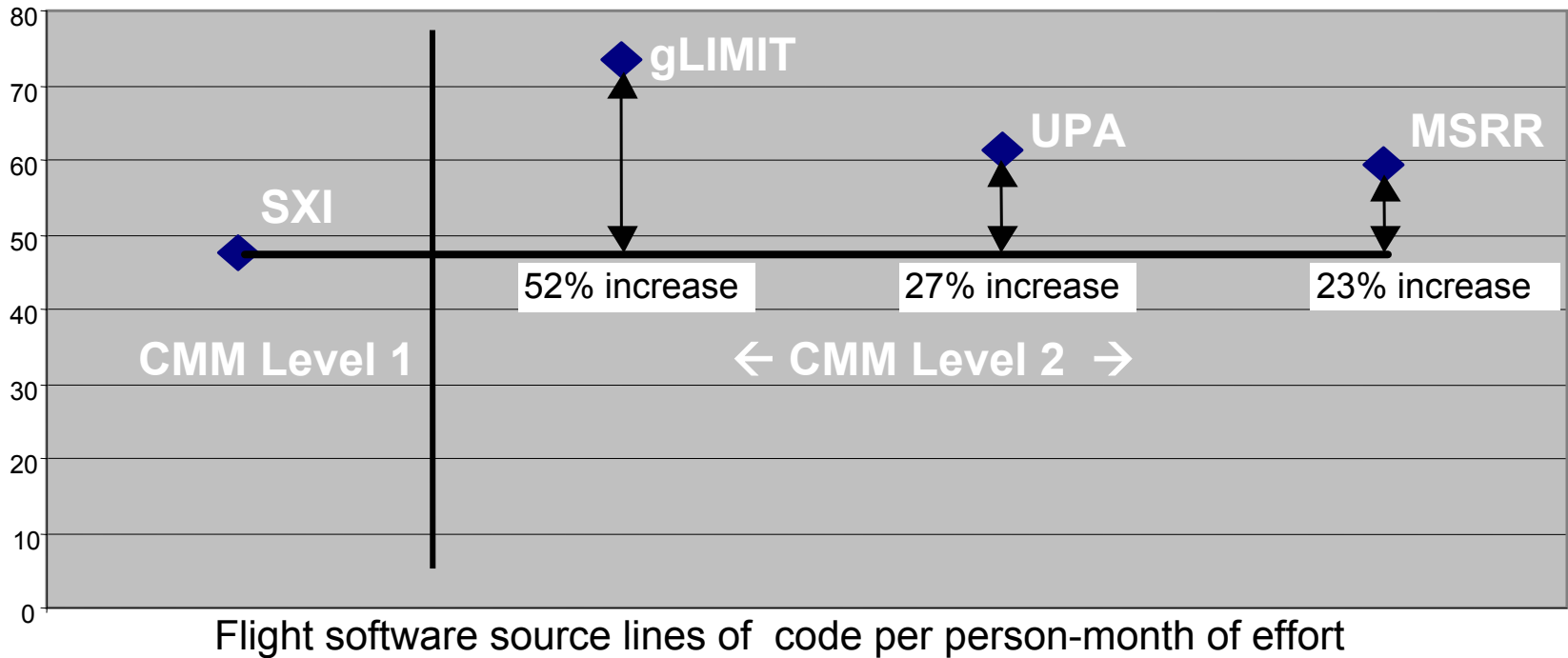


CMM Level	Percent Rework	Phase Containment Effectiveness	CRUD Density per KSLOC	Productivity (X Factor Relative)
2	23.2%	25.5%	3.20	1x
3	14.3%	41.5%	0.90	2x
4	9.5%	62.3%	0.22	1.9x
5	6.8%	87.3%	0.19	2.9x

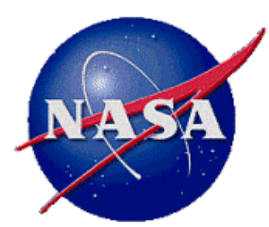
Diaz, M. & King, J., "How CMM Impacts Quality, Productivity, Rework, and the Bottom Line", Cross Talk: The Journal of Defense Software Engineering, March 2002. General Dynamics Decision Systems, 3 Divisions, 1,500 Engineers / 360 SW Engineers, CRUD = Customer Reported Unique Defects, Largest RIO found to be from levels 2 to 3 at 167% based on cost savings in rework.



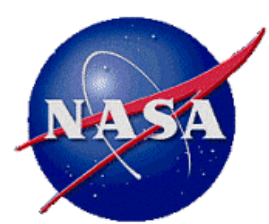
# Early Success on the NASA Software Initiative at MSFC: Reduced Cost



Software development productivity increased at Marshall Space Flight Center, the first Center to pilot SEI's Capability Maturity Model (CMM) in association with this Initiative



# NASA Improvement Initiatives



# NASA Systems Engineering Initiative



## Directed by NASA Chief Engineer:

“...the Software Engineering Working Group is expected to...define and pilot a methodology for assessment of the systems engineering capability, which addresses knowledge and skill of the workforce, processes, and tools and methodology.”

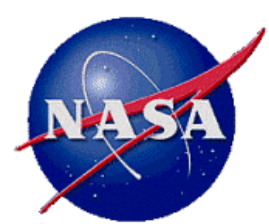
Deputy Chief Engineer for Systems Engineering (Nov. 1, 2000)

## Studied by NASA Systems Engineering Working Group (SEWG)

- Different assessment methods were evaluated by the SEWG to determine best methodology for benchmarking/improving Systems Engineering implementation agency-wide.
- Initial “quick-look” at systems engineering at GSFC using CMMI in 2002

## CMMI Pilot Appraisal at JPL in April 2004

- Did CMMI appraisal provide good benchmark of systems engineering capability?
- Was level of formality of CMMI appraisal used suitable for use at all Centers?



# NASA Software Engineering Initiative

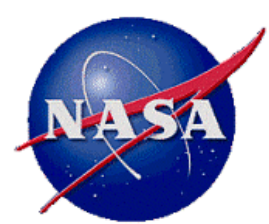
**Goal: *Advance software engineering practices (development, assurance, and management) to effectively meet the scientific and technological objectives of NASA.***

Strategy 1. Implement a continuous software process and product improvement program across NASA and its contract community.

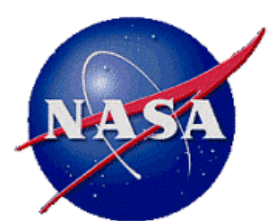
Strategy 2. Improve safety, reliability, and quality of software through the integration of sound software engineering principles and standards.

Strategy 3. Improve NASA's software engineering practices through research.

Strategy 4. Improve software engineers' knowledge and skills, and attract and retain software engineers.



# **GSFC Software Process Improvement**



# GSFC Software Process Improvement Plan



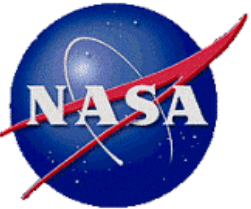
GSFC has a Software Process Improvement Plan, signed by Al Diaz, 9/01

Focus of Plan - **Improve the processes and practices in use at GSFC** using the Capability Maturity Model Integrated (CMMI) as a *measure* of progress

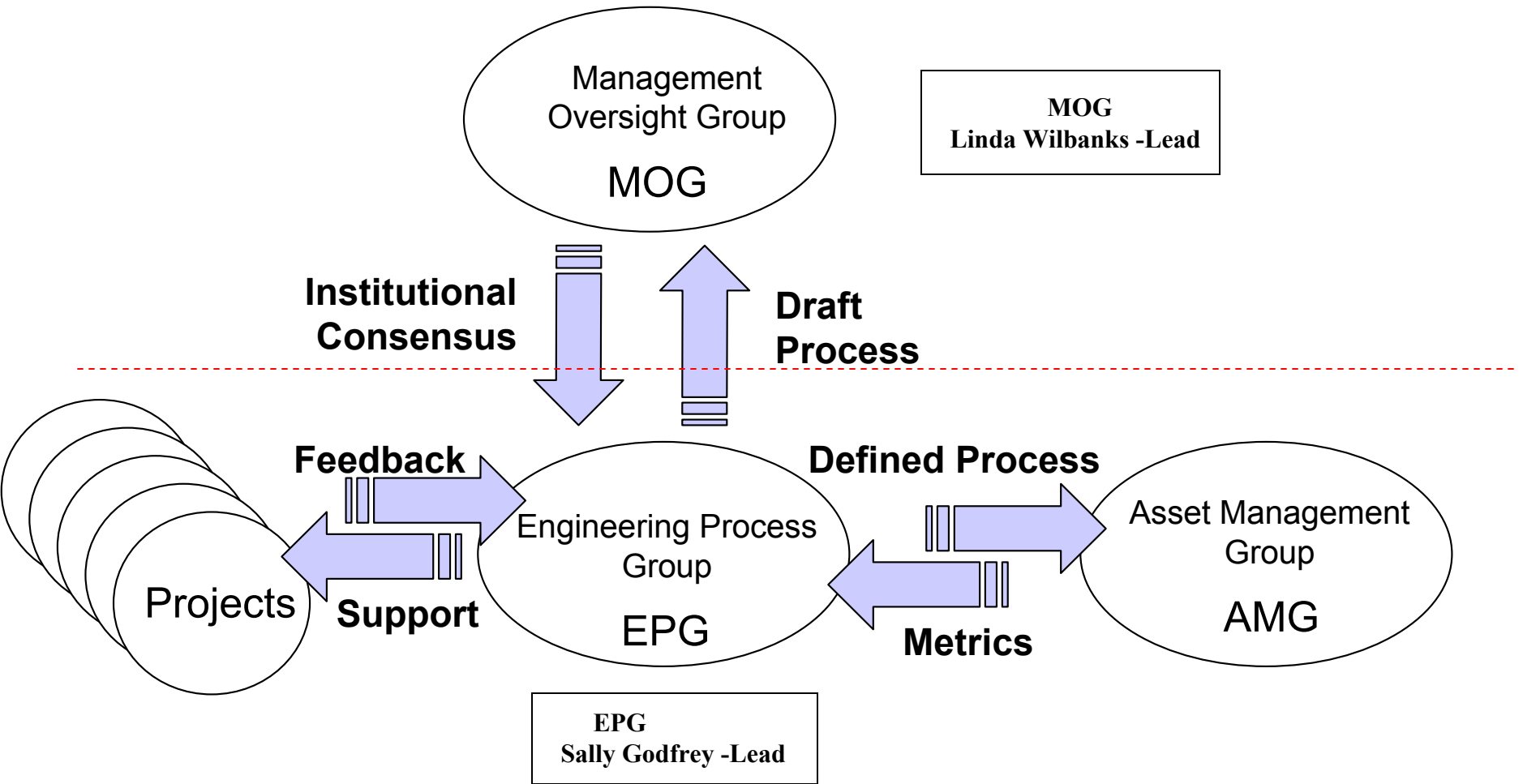
- GSFC Plan primarily addresses Strategy 1 in NASA Plan.
- FY04 Direction by Al Diaz: Achievement of specific CMMI goals

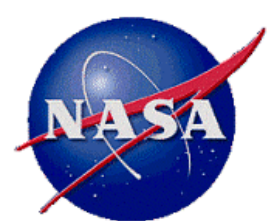
Do m a i n	F Y04	F Y05	F Y06	F Y07	F Y08
Fli ght S of tware B r a n c h		L e v e l 2	L e v e l 3		
I S D & C o d e 400 M i s s i o n S o f t w a r e			L e v e l 2	L e v e l 3	
A n y C o d e 600 /900 M i s s i o n S o f t w a r e n o t p r e v i o u s l y i n c l u d e d				L e v e l 2	L e v e l 3

Scope of Plan - All projects defined by NPG 7120.5 (Mission Software) & identified by Center Director will participate in this initial effort



# Infrastructure





# Implementation Phases in GSFC's Improvement Plan



## Phase 1: Pilot Phase (FY02)

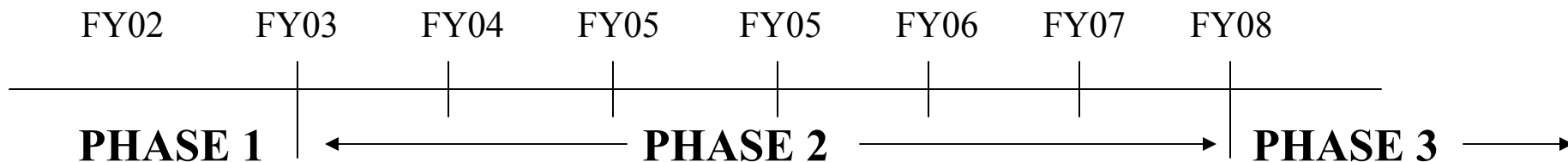
- Benchmark several representative GSFC areas
- Estimate effort, cost to improve identified gaps
- Evaluate implementation approach

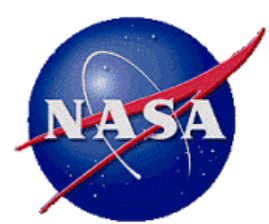
## Phase 2: Implementation Phase (FY03-FY08)

- Implementation of PI on all critical projects
- Begin by working with new projects to field improvements
- Target CMMI Level 3 for Mission Software

## Phase 3: Maintain Level and Continue Improvement

- Include other areas? (e.g. science processing)

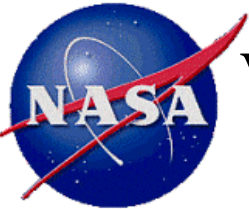




# GSFC Phase 1: Piloting FY02



- Conducted 3 sets of CMMI pre-appraisals
  - Appraisals were quick-look, Class B, C appraisals
  - Purpose of appraisals:
    - Evaluate use of CMMI, get better estimate of effort/ cost
    - Get a benchmark against CMMI model, identify gaps
- Sets of projects for pre-appraisals:
  - 2 flight software in-house led teams (included contractors)
  - 3 spacecraft projects (2 largely contracted, 1 in-house)
  - 2 ground support software in-house led teams
- CMMI appraisals identified a number of gaps that were independently identified
  - Actions from Code S/Y Colloquium produced a similar list
  - Plans for Phase 2 were based on findings from Phase 1



# What is broken (gaps) in the Agency's software engineering capability?



Centers are almost universally weak in:

## Project planning

Estimating cost, schedule, and resource requirements for project requirements fulfilled by software

## Monitoring and control of software engineering products

I.e., tracking progress and taking effective corrective actions

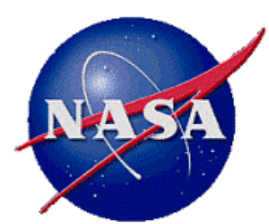
Configuration management is not universally applied throughout the software development process

Interface between software and system engineering processes is not well defined so agreements, audits, and reviews are not well planned or performed to achieve the most benefit

Software Quality Assurance is generally not well understood nor is its value appreciated

Findings by Raymond Kile, Authorized Lead Evaluator  
Center for Systems Management, Sept 2002

**GSFC's gaps were similar to findings across the Agency**



# GSFC Phase 2: Strategies (FY03-FY08)



- Use of CMMI SE/SW/SS **Continuous model**-- Early implementation of process areas that benefit us most
- Initial focus on **software** improvement --NASA Systems Engineering Working Group still determining direction
- First software area will be on **in-house flight software**, then **ISD/Code 400**
- Acquisition improvement activities begin in mid-FY04, gradual phase in
- Assets will be developed “**top-down/bottom-up**”
  - Top-Down: Define high level structure of documentation, training
  - Bottom-Up: Develop low level products for deployment, use FSW best practices to help develop high level process
- Phase in improvements on **newer projects**- Products developed as projects need them
- Project Plan updated for new CMMI goals - in signature cycle

Process Area Name	Abbr	ML	FY04	FY05	FY06	FY07	FY08
Requirements Management	REQM	2					
Measurement & Analysis	MA	2					
Project Monitoring & Control	PMC	2					
Project Planning	PP	2					
Process & Product Quality Assurance	PPQA	2					
Supplier Agreement Management	SAM	2					
Configuration Management	CM	2					
Decision Analysis & Resolution	DAR	3					
Product Integration	PI	3					
Requirements Development	RD	3					
Technical Solution	TS	3					
Validation	VAL	3					
Verification	VER	3					
Organizational Process Focus	OPF	3					
Organizational Process Definition	OPD	3					
Integrated Project Management	IPM	3					
Risk Management	RSKM	3					
Integrated Supplier Management	ISM	3					
Organizational Training	OT	3					

Key

Reviewed	
Capability Level 2	
Capability Level 3	

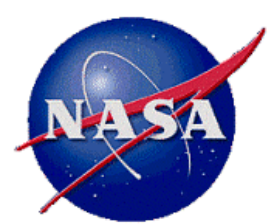
Table B-1: Process Area Schedule for Flight Software

Process Area Name	Abbr	ML	FY04	FY05	FY06	FY07	
Requirements Management	REQM	2					
Measurement & Analysis	MA	2					
Project Monitoring & Control	PMC	2					
Project Planning	PP	2					
Process & Product Quality Assurance	PPQA	2					
Supplier Agreement Management	SAM	2					
Configuration Management	CM	2					
Decision Analysis & Resolution	DAR	3					
Product Integration	PI	3					
Requirements Development	RD	3					
Technical Solution	TS	3					
Validation	VAL	3					
Verification	VER	3					
Organizational Process Focus	OPF	3					
Organizational Process Definition	OPD	3					
Integrated Project Management	IPM	3					
Risk Management	RSKM	3					
Integrated Supplier Management	ISM	3					
Organizational Training	OT	3					

Key

Reviewed	
Capability Level 2	
Capability Level 3	

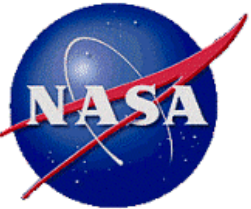
Table C-1: Process Area Schedule for ISD/Code 400 Mission Software



# GSFC Phase 2: Focus Activities Beginning FY03



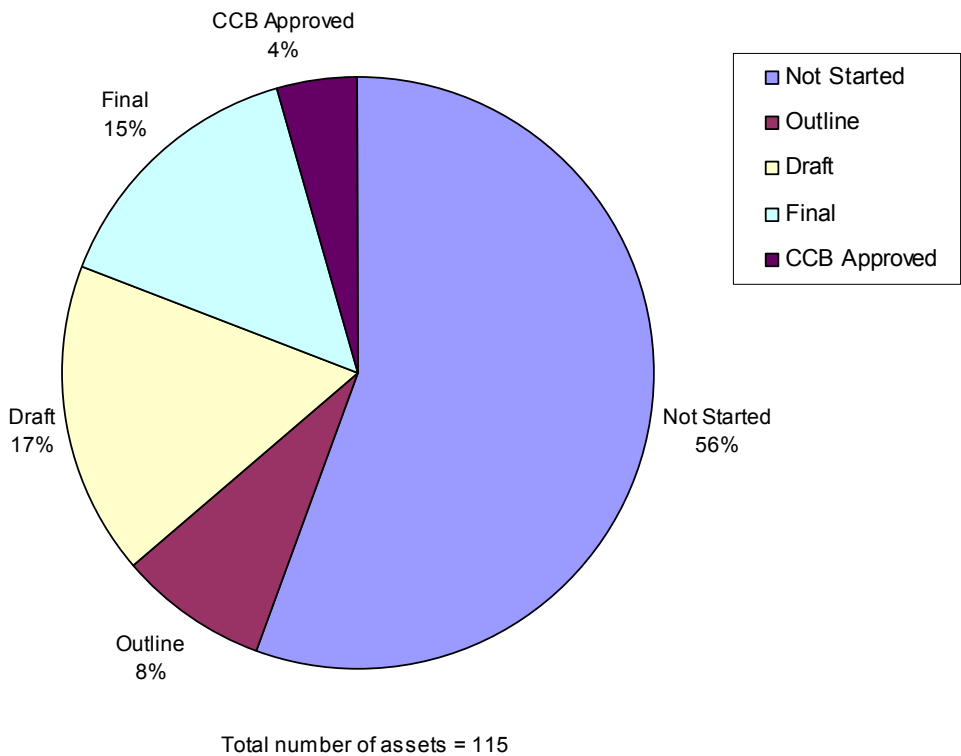
- **Code 582:** Flight Software:
  - Documentation of existing best practices (& suggested improvements)
  - Tools, checklist, templates to support consistent use of practices (e.g. requirements inspection procedures, test plan/procedure templates)
  - Training to support use of improved practices
  - Identification & support for collection/analysis of measures
- **Code 580:** Using flight software practices as a basis, best practices will be documented for all of ISD with assoc. work products & training
  - Consistent approach to planning and tracking (WBS, Earned value, Risk Management)
- **Code 590:** Have worked with NASA systems engineering group to pilot use of CMMI for systems engineering appraisals (JPL was first pilot)
- **Code 400:** Software Acquisition improvements beginning with developing improved RFP templates for software - Review at JPL/GSFC QMSW workshop
- **Code 300:** Began improvements in Software Assurance



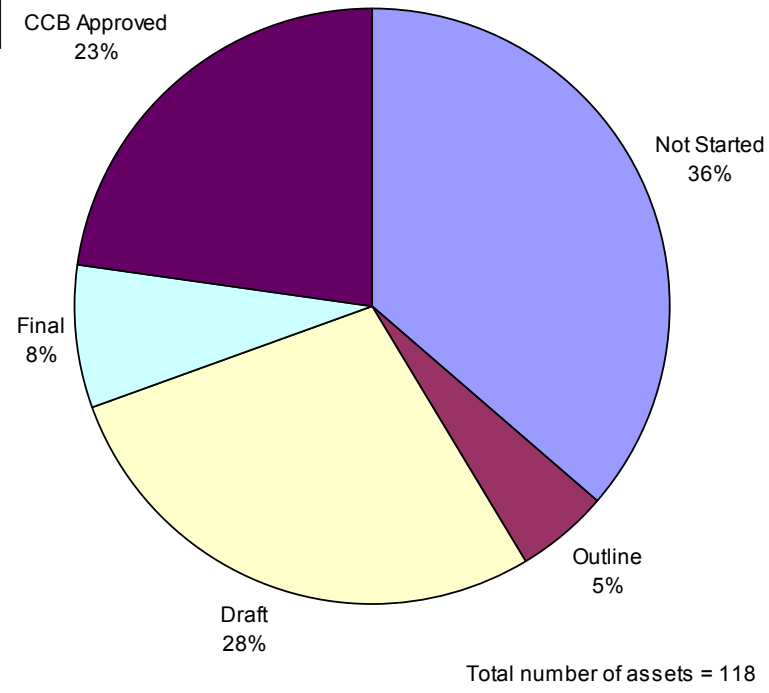
# Summary-Process Documentation Development Progress (FSW & ISD) as of April 13, 2004

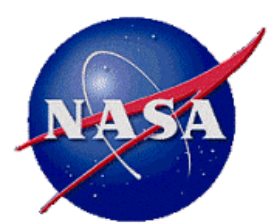


**Status of ISD Process Assets**



**Status of Tailored FSW Process Assets**



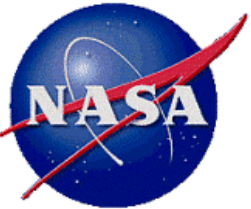


# Overall Concepts- Documentation



- Will be a “generic” set of procedures/processes for ISD/GSFC
- “Generic set” will be tailored for Branches (FSW) or classes of software (e.g.-ground systems, science processing, research...) Must use Tailoring Guidelines.
- Projects can also tailor, based on tailoring guidelines
- ISD/GSFC documentation will be on EPG web site
  - Branch tailored documentation can be on Branch web sites
  - Web sites will include use-aids: checklists, templates
- Training and tools will be available with processes

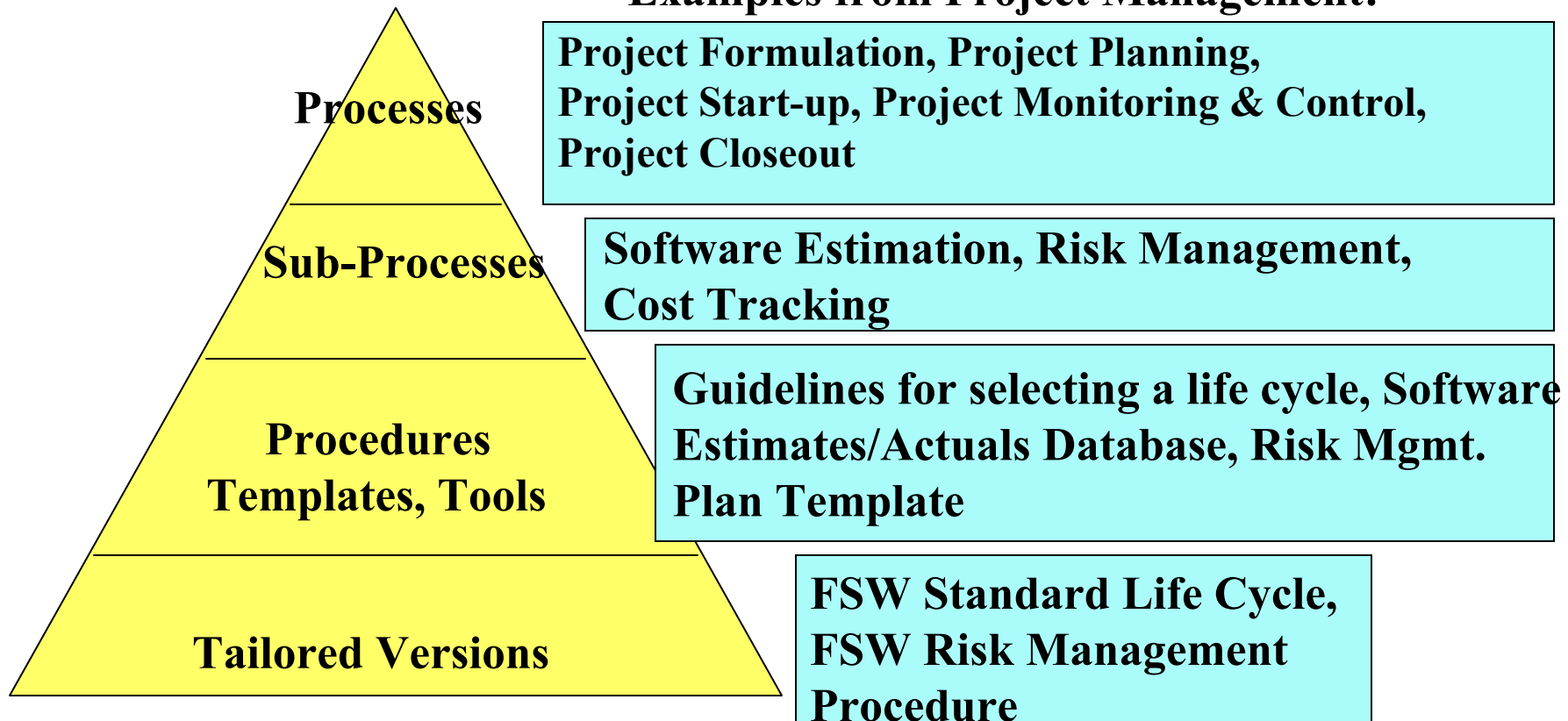




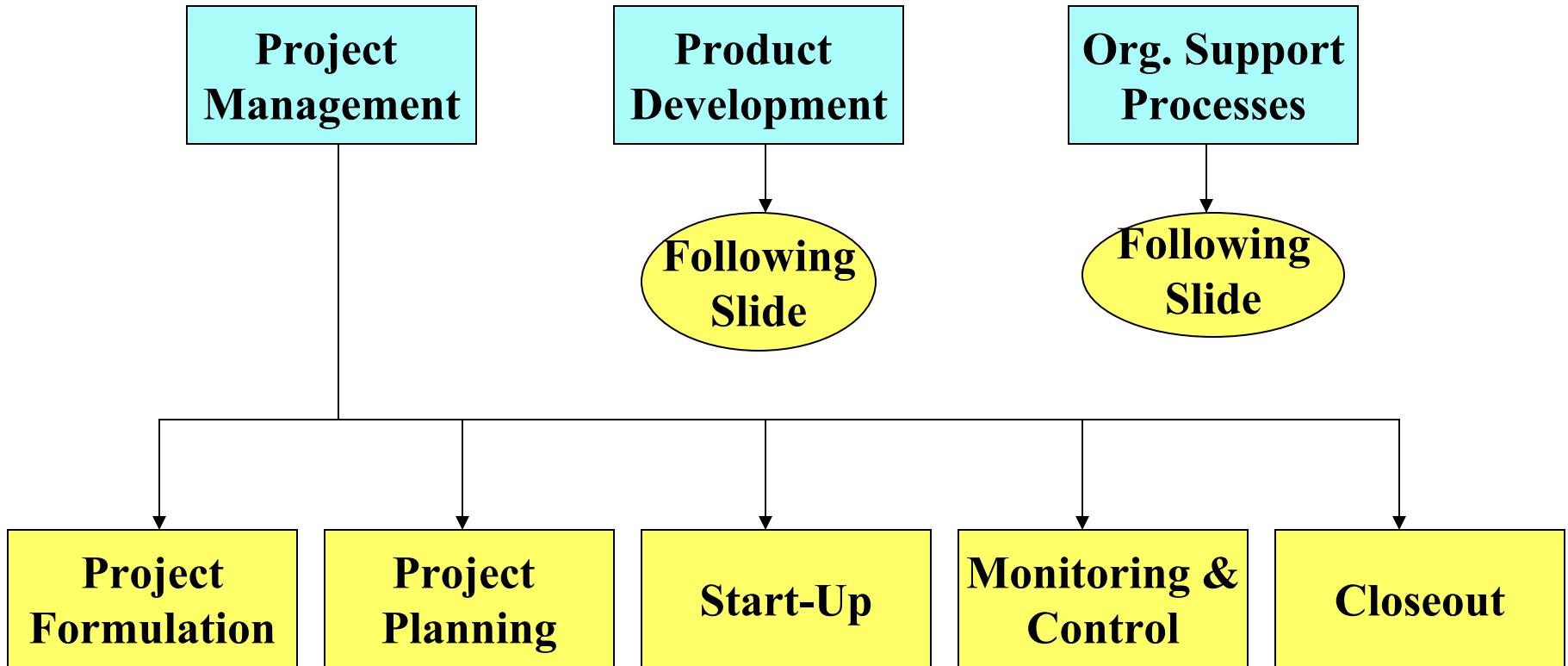
# Process Documentation Structure-Top-Down View

**Documentation is divided into three Process categories:  
Project Management Processes, Product Development Processes,  
Organizational Support Processes**

**Examples from Project Management:**



# Description of Processes to be Documented



# Description of Processes to be Documented

**Project  
Management**

**Previous  
Slide**

**Product  
Development**

**Org. Support  
Processes**

**Following  
Slide**

**Systems  
Engineering**

**Requirements  
Engineering**

**Design**

**Implementation**

**Testing**

**Product  
Release**

**Sustaining Eng.  
& Maint.**

# Description of Processes to be Documented

**Project  
Management**

**Previous  
Slide**

**Product  
Development**

**Previous  
Slide**

**Org. Support  
Processes**

**Configuration  
Management**

**Quality  
Assurance**

**Training**

**Measurement  
& Analysis**

**Process  
Engineering**





GSFC SW Improvement	Process Assets Library (PAL)	Training	Measurement	Lessons Learned
------------------------	---------------------------------	----------	-------------	-----------------

## Process Asset Library

- [+About the PAL](#)
- [+PAL Feedback Form](#)
- [+PAL Help](#)
- [+Glossary](#)

## PAL Contents

- [+Project Management](#)
- [+Product Development](#)
- [+Organizational Support](#)
- [+PAL Index](#)
- [+Assets by Role](#)
- [+Assets by Tailoring](#)
- [+Assets by Type](#)
- [+Policies](#)
- [+Standards](#)

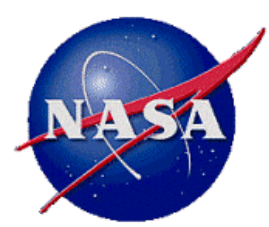
## Welcome to the GSFC Process Assets Library

The Process Assets Library (PAL) is the repository for all process assets that have been approved for software development at GSFC. Assets include policy, procedures, process descriptions, document templates, guidelines, standards, checklists, and tools.

The initial set of assets has been developed for ISD, but will ultimately be augmented to serve all GSFC projects.

PAL assets may be assessed in multiple ways. The following table shows how these access routes, or “views” can help you find the assets you need.

View	What the view provides
Contents	A table of contents for the PAL
Index	An alphabetical index into the PAL
Role	A list of the roles of personnel working on a typical software project, showing the process assets needed by each role and training courses for each role
Tailored	A set of process assets that have been created or “tailored for use on a specific project or in a specific domain
Description	High level descriptions of the 3 asset categories & the processes they contain
Asset Type	A set of all assets of the same type; e.g., all “templates” or all “checklists”

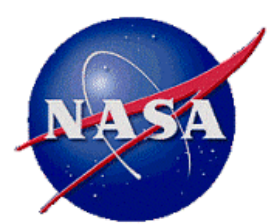


# Features of Software Training Web Page



Training Page Includes:

- Training Program Information
  - Software Classes Calendar & GSFC Training Calendar
  - Role Based Training Matrix
  - On-line Training (self-paced, presentations, etc)
  - Software Certification Information
- Software Conference Information
- “Ask an Expert” Feature
- Training Support Page
  - Help in Developing a Class (Can request new class)
  - Mentoring Information
  - How to schedule a class, Feedback on Classes
- Other Training Links



# Other Features of Software Web Site

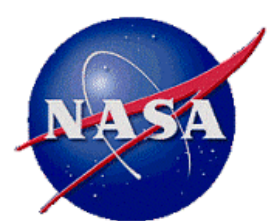


## Lessons Learned web page features:

- “Submit a Lesson”
- Software-Specific Lessons Learned Library with views by roles, categories, phases
- Subscribe/Unsubscribe Features
- Lessons Learned Feedback
- Link to “Experts”
- Questions and Answer Forum

## Measurement Repository web site features:

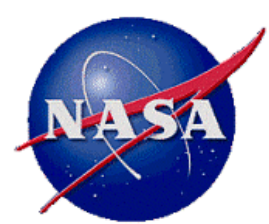
- On-line submission of measures
- Access to Measurement Database (for authorized users)
- Measurement Analysis and Charts
- Guidance in establishing and measurement programs



# Software Training Associated with Process Improvement



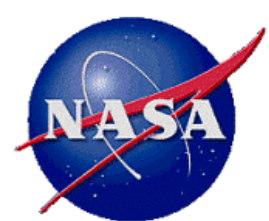
Audience	Focus	Approach
Community/Others Interested	General Awareness	<ul style="list-style-type: none"><li>-Overview info on CMMI, improvement initiative</li><li>-Lectures, teas, overview classes</li></ul>
Developers/Team Leads	ISD/GSFC specific practices	<ul style="list-style-type: none"><li>-Role-based approach</li><li>-Train on documented procedures, guidelines, templates</li></ul>
Developers/Team Leads	Discipline expertise	<ul style="list-style-type: none"><li>-Focus on general skills</li><li>-University classes, 3rd party classes, teas, conferences</li></ul>
Software Customers	Products, Software/ Customer Interface	<ul style="list-style-type: none"><li>-Emphasis on products delivered &amp; needs for producing products</li><li>-Use of products</li></ul>



# Progress Highlights in FY03/FY04



- Flight Software:
  - FSW “Standards” CCB; 27 products baselined and available
  - Are developing products “in-time” to meet project needs
  - Products in use on all new FSW projects, some existing
- ISD/Code 400:
  - Have ISD CCB for processes; 7 products baselined and available
  - Have developed templates for software parts of RFP’s
  - Have developed a class to help project managers manage software
  - Have sponsored classes in inspections, software configuration management, software safety, software acquisition, quantitative project management
- Code 300:
  - Have developed processes and checklists
  - Training for better software assurance



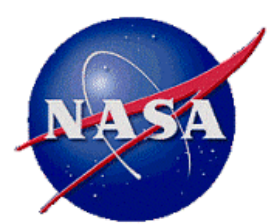
# Plans for FY04/FY05



- First pre-appraisal in mid-August on Flight Software: Plan to **look at** (gap analysis):

Project Planning	Project Monitoring & Control
Requirements Management	Requirements Development
Configuration Management	Software Assurance
Risk Management	Organizational Process Focus

- Target SCAMPI (formal appraisal) in October for a few process areas
- Rest of level 2 processes for FSW in FY05, some of level 3 processes
- Will phase in level 2 processes for ISD ASAP, target capability level 2 appraisal in FY05

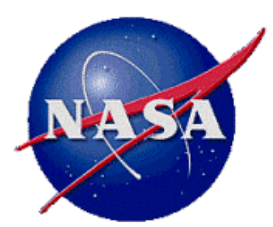


# Summary



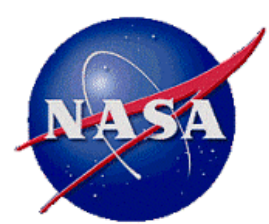
- GSFC is moving forward to improve our software processes and products using CMMI as an improvement model
  - Phase 1 identified many potential areas for improvements
  - Phase 2 work has started work in a variety of areas and is beginning to deploy software improvements
  - We are working towards achievement of CMMI Level 2 in a few process areas by early FY05 and CMMI Level 3 by late FY07
  - We hope to coordinate with systems engineering improvements

**“Better Software/Systems Engineering to Support Our Projects”**



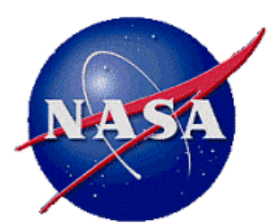
---

# Back-up Slides



# What Now?

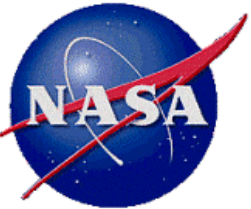
- GSFC Software Improvement Site: <http://software.gsfc.nasa.gov>
- For CMMI model reference go to:  
<http://www.sei.cmu.edu/cmmi/products/models.html>
- Can Download CMMI-SE/SW(IPPD)/SS V1.1 Staged
- Attend a CMMI Overview class or an Introduction to CMMI class for more details
- **What you really need to know is what processes you should be using to do your job well**
  - Define and use a good process
  - Measure against the CMMI model
  - Improve your process



# CMMI and ISO

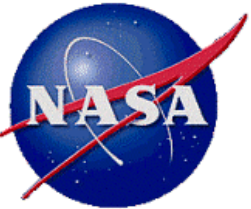


- **ISO is a standard, CMMI is a model**
  - **ISO is broad- focusing on more aspects of the business. Initially for manufacturing**
  - **CMMI is “deep”- provides more in-depth guidance in more focused areas (Software/Systems Engineering/Software Acquisition-SW/SE/SA)**
  - **Both tell you “what” to do, but not “how” to do it**
  - **But CMMI tells you what “expected” practices are for a capable, mature organization**
- 
- **CMMI provides much more detail for guidance than ISO by including an extensive set of “best practices”, developed in collaboration with industry/gov/SEI**
    - CMMI provides much better measure of quality of processes; ISO focuses more on having processes
    - CMMI puts more emphasis on continuous improvement
    - CMMI allows you to focus on one or a few process areas for improvement (It’s a model, not a standard, like ISO) --Can rate just one area in CMMI
    - CMMI and ISO are not in conflict: ISO helps satisfy CMMI capabilities; CMMI more rigorous



# What is CMMI? What do levels of software engineering maturity mean?

Level	Description	Process Areas	Result
Optimizing 5	Improvement institutionalized- routinely fed back into the process	Causal Analysis & Resolution Organizational Innovation & Deployment	<b>High Productivity &amp; Quality</b>
Quantitatively Managed 4	Product and process are quantitatively controlled	Organizational Process Performance Quantitative Project Management	
Defined 3	Software engineering and management processes defined and integrated - processes standardized	Organizational Process Focus Organization Process Definition Organizational Training Integrated Project Management Technical Solution/Product Integration Integrated Supplier Management Verification/ Validation Risk Management Decision Analysis Resolution	
Managed 2	Basic project management in place; performance is repeatable	Requirements Management Project Planning Project Monitoring and Control/ Supplier Agreement Management Process & Product Quality Assurance Configuration Management Measurement & Analysis	
Initial 1	Ad Hoc	Processes are informal and unpredictable	<b>High Risk</b>

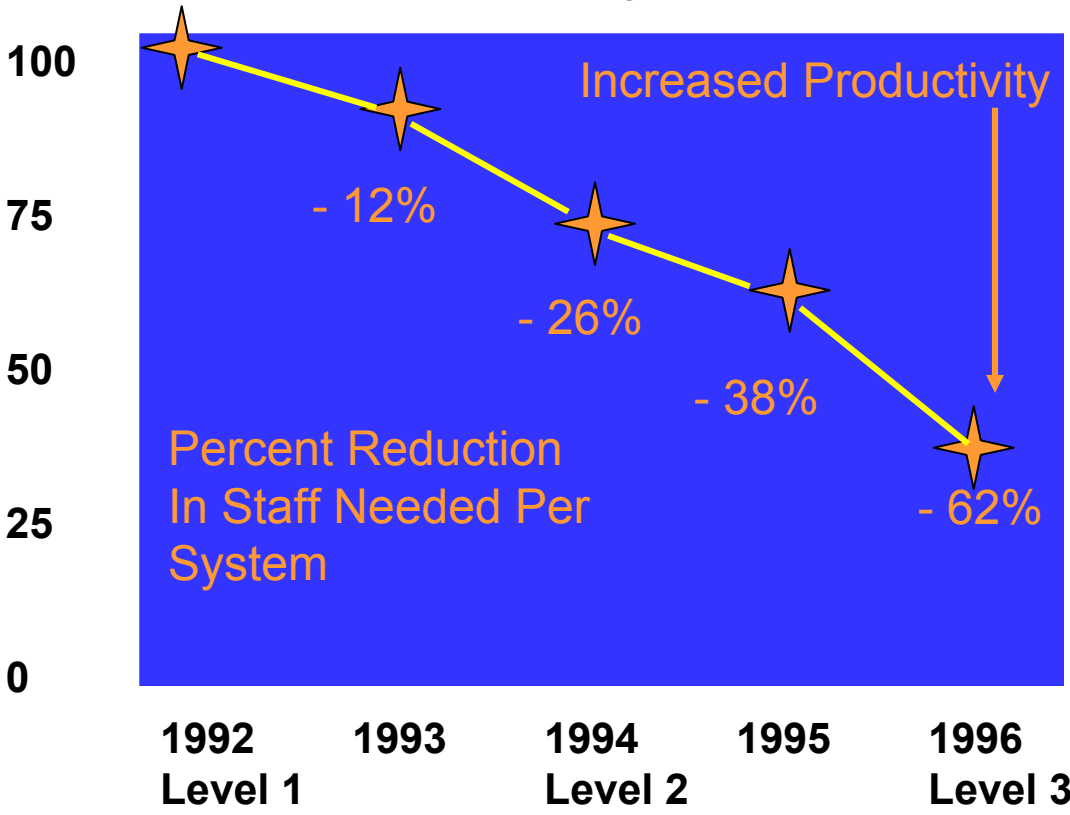


# Time History - Productivity



## Productivity

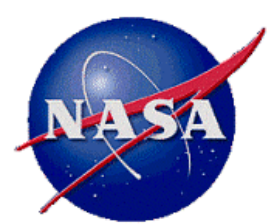
Reduced Staff Support Per System = Increased Productivity



**Projects at Maturity Level 3 Increased Productivity 62%**

Based on 120 Projects at Boeing Information Systems

Reference: Scott Griffin, Chief Information Officer, The Boeing Company, SEPG Conference, 2000.

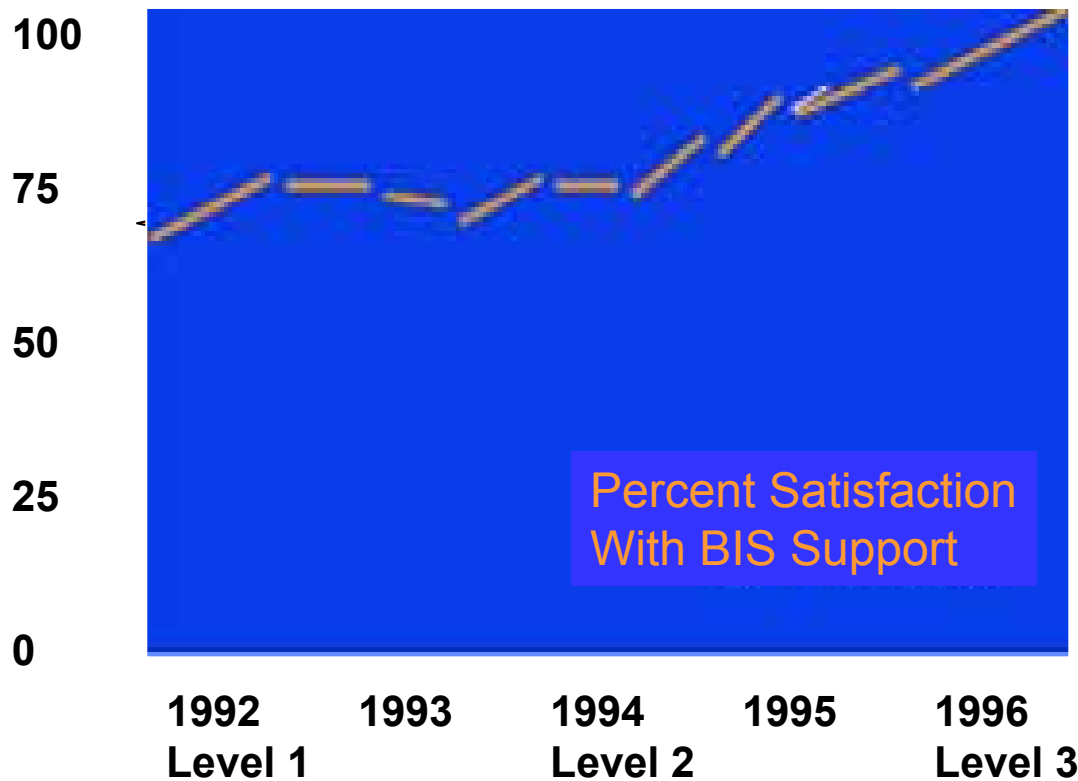


# Time History – Satisfaction



## Customer Satisfaction

Based on Semi-Annual Survey of Customers

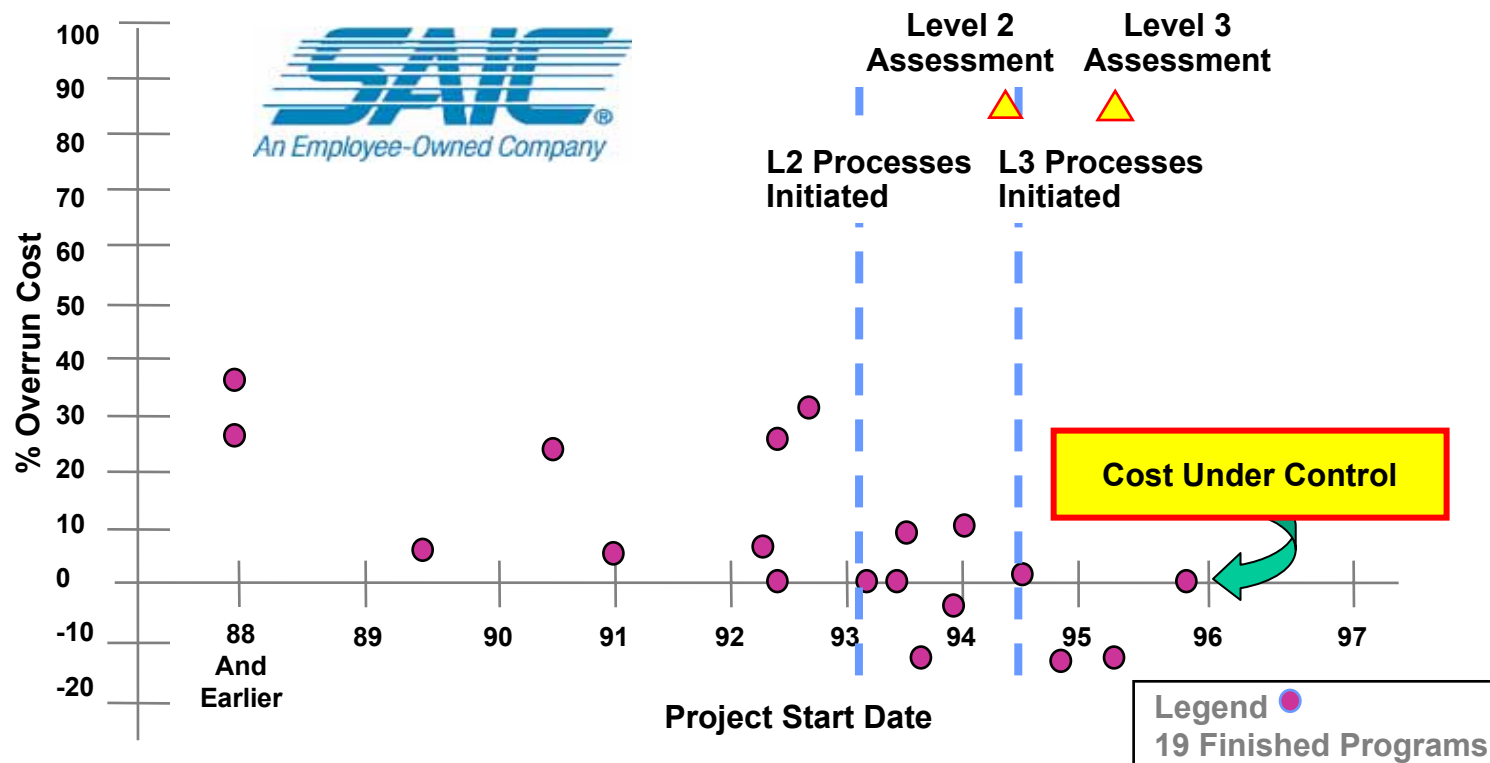


**Customer Satisfaction Increased with CMM Level**

**Based on 3 Major Programs in Boeing Defense and Space Group**

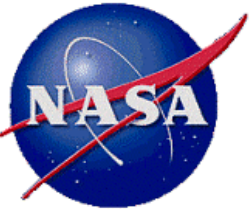
Reference: Scott Griffin, Chief Information Officer, The Boeing Company, SEPG Conference, 2000.

Task 4 WBS 3.6.5

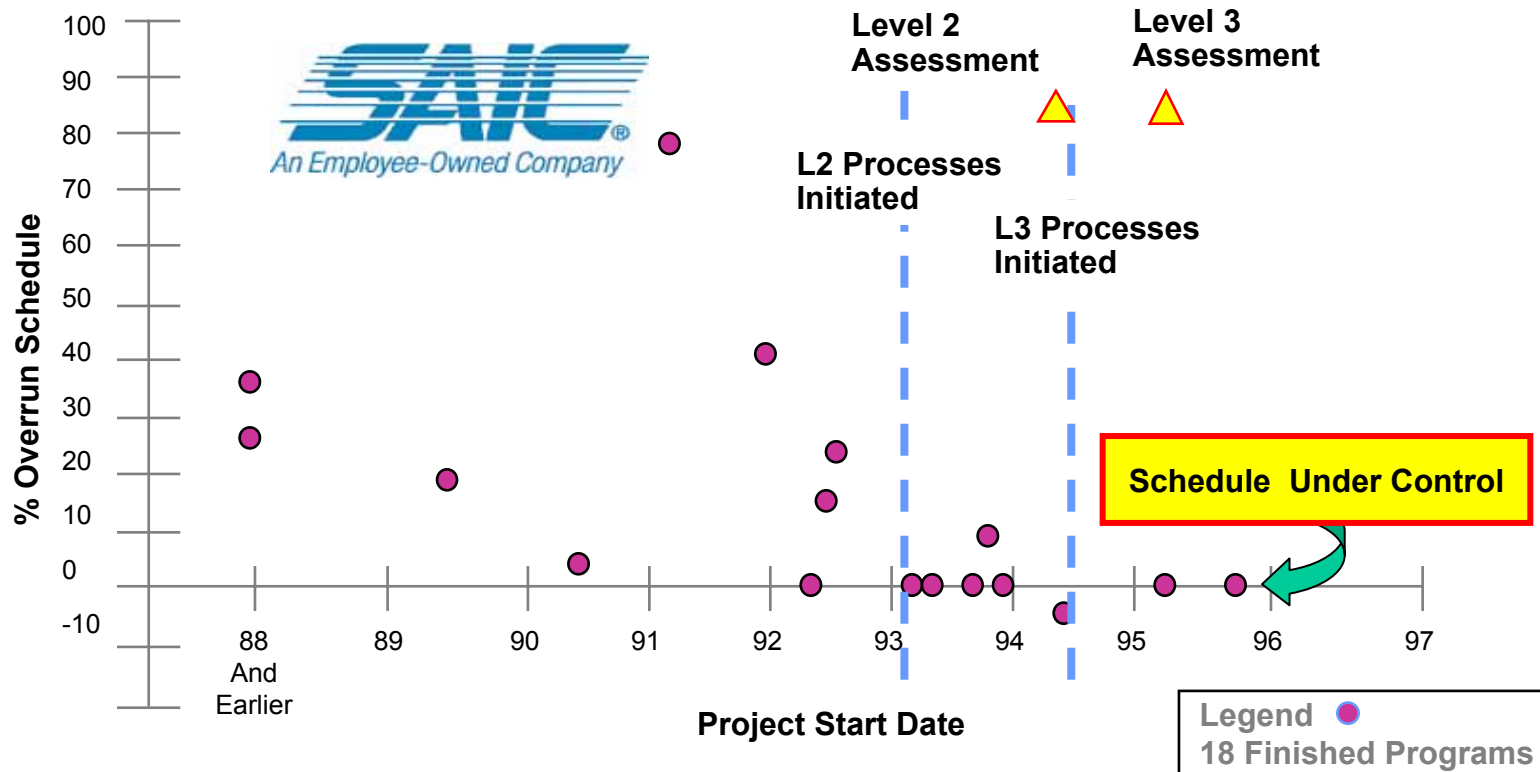


Source: Software-related engineering projects completed for SAIC Aeronautical Systems Operation during 1984 -1996 for all contract types and contract size \$80K to \$3.5M.

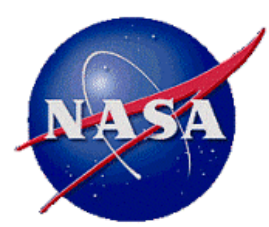
## Task 4 WBS 3.6.5



# Time History – Schedule



Source: Software-related engineering projects completed for SAIC Aeronautical Systems Operation during 1984 -1996 for all contract types and contract size \$80K to \$3.5M.



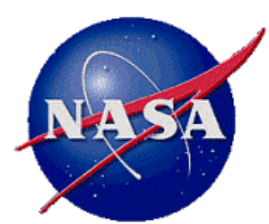
# Even Successful Missions experience software problems

“A few days after the July 4th, 1997 landing, the Mars Pathfinder began experiencing total system resets, each resulting in losses of data. The problem was a logical error in the real-time scheduling system---a classic priority-inversion problem. Fortunately, this problem was repairable from earth.

A malfunction in one of the on-board computers on Clementine on May 7, 1994 caused a thruster to fire until it had used up all of its fuel, leaving the spacecraft spinning at about 80 RPM with no spin control. This made the planned continuation of the mission, a flyby of the near-Earth asteroid Geographos, impossible.

The Magellan spacecraft broke Earth lock and lost communications several times in August 1990 (soon after entering Venus orbit). It took over six months to identify the source of the problem, which was a timing error in the flight software.”

*- Ricky Butler, NASA Langley's Formal Methods Research Program Overview*



# Launch Failures Caused by Design Errors



- “The April 30, 1999 loss of a Titan I, which cost the taxpayers \$1.23-billion, was due to incorrect software (incorrectly entered roll rate filter constant)
- Aug 27, 1998 failure of the Boeing Delta 3 launch vehicle (control system attempted to correct a roll oscillation and the hydraulic fluid used to move the nozzles on the solid-rocket motors with TVCs was depleted. )
- On 4 June 1996, the maiden flight of the Ariane 5 launcher exploded (a software exception was caused during a data conversion )”  
*“Three successive Titan IV mission failures, an Athena failure and two straight mission losses of the large new commercial Delta III, including its latest mishap May 4, mark the worst string of major U.S. launch accidents in 13 years.”*

- Ricky Butler, NASA Langley's Formal Methods Research  
Program Overview